

ОБ ОДНОМ АЛГОРИТМЕ РАСЧЕТА КАЛЕНДАРНОГО ПЛАНА
ЗАПУСКА-ВЫПУСКАЮ. Н. ЕФИМОВ, В. И. КИЗЕВ, В. И. НЕВРАЕВ, П. А. СЕДЕЛЬНИКОВ

(Представлена научным семинаром УВЛ)

В работе [1] предлагается алгоритм расчета календарного плана запуска-выпуска, использующий сетевую модель и минимизирующий незавершенное производство. Практическое внедрение разработанной системы планирования показало, что первая попытка реализации этого алгоритма, описанная в той же работе, не совсем удачна.

Представление плана выпуска для каждой вершины графа *) в виде множества

$$P_j^B(t) = \{q_\lambda^B, t_\lambda^B\}_{\lambda=\overline{1, \mu}; j \in I}$$

является весьма громоздким. Это приводит к большим объемам полей для хранения планов, нерациональному использованию НФ и НМЛ, что, в свою очередь, ведет к значительным затратам машинного времени анализа сетевой модели. Неодинаковое количество элементов в планах различных вершин графа усложнило сервисную программу и потребовало введения таблицы информации (ТИ), хранимой в НФ.

Экспериментально установлена степенная зависимость времени расчета от количества вершин анализируемой модели. Накопленный опыт использования этого алгоритма для календарного планирования на ЭВМ «Урал-11Б» показал, что на машинах подобного класса могут быть просчитаны сетевые модели, не превышающие 1000 вершин. Необходимость анализа графов большей размерности требует разработки другого машинного алгоритма расчета планов запуска-выпуска, свободного от вышеуказанных недостатков. Нам представляется, что решение поставленной задачи нужно искать на пути более компактной записи планов и разработки более совершенной системы их хранения и поиска в памяти машины.

Исходя из вышеизложенного, нами предлагается другой вариант алгоритма расчета планов, отличающийся от [1] как формой задания графа, так и методикой анализа.

Сетевая модель представляется в виде графа $G = (I, \Gamma)$, так что информация об одной вершине имеет вид

$$\begin{aligned} & i, x_i, T_i, y_i, m_i, \rho_i; \\ & j_1, R_{i,1}; \\ & j_2, R_{i,2}; \\ & \dots \end{aligned} \tag{1}$$

*) Здесь и в дальнейшем терминология и обозначения взяты из работы [1].

$$j_p, R_{i,p},$$

где $j_k \in \Gamma^i$, причем $i > j$, а $j_k < j_{k+1}$;

ρ_i — число дуг, исходящих из вершины i (локальная степень G в i). Отметим, что такую информацию легко получить из сводной спецификации на изделие. В случае, если информация берется из угловых спецификаций или из технологической документации (ведомостей комплектации), нетрудно ее преобразовать к виду (1), используя алгоритм [2]. Причем исходная информация должна быть упорядочена по слоям, а внутри слоя по возрастанию номеров вершин i . Упорядочение осуществляется алгоритмом [3].

Планы выпуска формируются в виде конечного упорядоченного множества.

$$P^B j(t) = \{q_\lambda^B\}_{\lambda=\overline{1,v}},$$

где v — количество элементов плана. Величина $q_\lambda^B \geq 0$ характеризует выпускаемое количество, а ее местоположение (позиция) в упорядоченном множестве $P_j^B(t)$ — время выпуска. Необходимо подчеркнуть, что в отличие от [1] здесь может быть $q_\lambda^B = 0$, что и позволяет применить позиционную форму записи планов. При такой форме записи для представления одного плана выпуска требуется не более Z^B ячеек оперативной памяти ЭВМ.

$$Z^B = \frac{T^n + t_u}{h} \cdot \psi_q,$$

где T^n — плановый период;

t_u — длительность производственного цикла изготовления выпускаемых изделий;

h — шаг квантования функции директивного плана;

ψ_q — отношение количества разрядов, необходимых для записи q_λ^B , к количеству разрядов одной ячейки НФ.

Аналогично планы запуска представляются в следующей позиционной форме:

$$P_j^3(t) = \{n_\lambda\}_{\lambda=\overline{1,v}}, \quad (2)$$

где n_λ — количество партий m_j , запускаемых в момент времени, определяемый позицией n_λ в $P_j^3(t)$. При этом под каждый план запуска требуется Z^3 ячеек оперативной памяти.

$$Z^3 = \frac{T^n + t_u}{h} \cdot \psi_n,$$

где ψ_n — отношение количества разрядов, необходимых для записи величины n_λ , к количеству разрядов одной ячейки НФ.

Практически для большинства производств запуск изделий производится по одной партии m_j . В этом случае в каждый фиксированный момент времени в $P_j^3(t)$ отмечается только сам факт запуска, для чего используется один двоичный разряд. Такая форма записи планов запуска является достаточно компактной, причем ее можно сделать постоянной длины.

Работа предлагаемого алгоритма заключается в следующем: сначала по директивным планам для выпускаемых изделий формируются планы запуска. Затем просматривается упорядоченная исходная информация и для каждого $i \in \Gamma_j^{-1}$ формируется план выпуска по формуле:

$$P_i^B(t) = \sum_{j \in \Gamma_i} P_j^3(t) \cdot R_{ij}.$$

Полученный план выпуска преобразуется в план запуска этой же вершины согласно алгоритму [1], но представляется в машине в форме (2). Преобразованный план выпуска в дальнейших расчетах не участвует и в памяти не хранится. Таким образом, в результате работы алгоритма сохраняются только планы запуска для всех $j \in I$. Для работы алгоритма необходимо, чтобы к моменту расчета $P_i^B(t)$ планы $P_j^3(t)$ для всех $j \in \Gamma_i$ были уже рассчитаны. Этому требованию удовлетворяет упорядоченная исходная информация.

Записи планов запуска сделаны постоянной длины, что упростило их поиск в НФ и НМЛ. В начале процесса счета планы запуска записываются в НФ в поле планов до его заполнения, после чего планы пишутся в рабочее поле НМЛ. В дальнейшем в рабочее поле НМЛ переписываются только вновь получаемые планы запуска, так как они не потребуются для формирования планов всех остальных вершин анализируемого слоя. При переходе к анализу очередного слоя графа G планы из рабочих полей НФ и НМЛ переписываются в поле планов НМЛ, причем в НФ оставляются планы, полученные в последнюю очередь. Количество этих планов определяется размером поля планов НФ. В специальной ячейке оперативной памяти формируются верхняя i_v и нижняя i_n границы подмножества $I_{\text{нф}} \subset I$ вершин графа G , планы которых находятся в НФ. Тогда поиск необходимых для расчета планов можно осуществить следующим образом:

а) Если $j \in I_{\text{нф}}$, то $A_j^{\text{н}} = A_0 + (j - i_n) \cdot Z^3$,

где $A_j^{\text{н}}$ — начальный адрес записи плана $P_j^3(t)$;
 A_0 — начальный адрес поля планов.

б) Если $j \notin I_{\text{нф}}$, то $N_z = N_z^0 + E' \left(\frac{j}{\Theta} \right)$,

где N_z — номер зоны НМЛ, в которой записан искомый план;
 N_z^0 — начальный номер зоны поля планов НМЛ;

Θ — количество планов запуска, хранящихся в одной зоне;

$E' \left(\frac{j}{\Theta} \right)$ — целая часть отношения. Остаток этого отношения опреде-

ляет местоположение плана внутри зоны.

Таким образом, в результате работы алгоритма получаем в НМЛ планы запуска для всего множества вершин анализируемой сетевой модели. Эта информация служит основой для формирования на ЭВМ следующих выходных документов:

календарного плана запуска-выпуска, разнесенного по подразделениям; сводного плана выпуска в трудовом и стоимостном выражении в целом по изделию и разнесенного по подразделениям; плановой потребности в основных и вспомогательных материалах, комплектующих, инструменте, энергозатратах и т. д. в натуральном и стоимостном выражении; различных лимитных карт по подразделениям.

Предложенный алгоритм был реализован на ЭВМ «Урал-11Б». Время анализа сетевой модели, содержащей 4000 вершин при коэффициенте сложности 3, составило примерно 30 мин.

ЛИТЕРАТУРА

1. Ю. Н. Ефимов, В. И. Кизев, В. И. Невраев, П. А. Седельников. Алгоритм решения задачи оперативного календарного планирования дискретного производства. Настоящий сборник.
2. Ю. Н. Ефимов, П. А. Седельников. Об одном алгоритме преобразования формы представления графа. Изв. ТПИ, т. 243. Изд. ТГУ, 1972.
3. Ю. Н. Ефимов, П. А. Седельников. Алгоритм упорядоченной перекодировки вершин графа. Настоящий сборник.